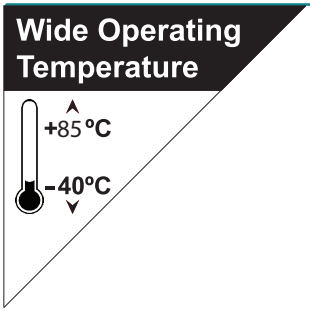


ETX-742E

Wide Range Temperature
Intel® Atom™ N450 ETX® CPU module

User's Manual

Version 1.1

2014.08

This page is intentionally left blank.

Table of Contents

Chapter 1

Introduction

1.1 Copyright Notice

The information in this document is subject to change without prior notice in order to improve the reliability, design and function. It does not represent a commitment on the part of the manufacturer.
Under no circumstances will the manufacturer be liable for any direct, indirect, special, incidental, or consequential damages arising from the use or inability to use the product or documentation, even if advised of the possibility of such damages.

1.2 About This User's Manual

This User's Manual is intended for experienced users and integrators with hardware knowledge of personal computers. If you are not sure about any description in this User's Manual, please consult your vendor before further handling.

1.3 Warning

Single Board Computers and their components contain very delicate Integrated Circuits (IC). To protect the Single Board Computer and its components against damage from static electricity, you should always follow the following precautions when handling it :

1. Disconnect your Single Board Computer from the power source when you want to work on the inside.
2. Hold the board by the edges and try not to touch the IC chips, leads or circuitry.
3. Use a grounded wrist strap when handling computer components.
4. Place components on a grounded antistatic pad or on the bag that came with the Single Board Computer, whenever components are separated from the system.

1.4 Replacing the lithium battery

Incorrect replacement of the lithium battery may lead to a risk of explosion. The lithium battery must be replaced with an identical battery or a battery type recommended by the manufacturer.
Do not throw lithium batteries into the trashcan. It must be disposed of in accordance with local regulations concerning special waste.

1.5 Technical Support

If you have any technical difficulties, please do not hesitate to call or e-mail our customer service.

http://www.arbor.com.tw
E-mail:info@arbor.com.tw

1.6 Warranty

This product is warranted to be in good working order for a period of two years from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster.

Vendor assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, or inability to use this product. Vendor will not be liable for any claim made by any other related party.
Vendors disclaim all other warranties, either expressed or implied, including but not limited to implied warranties of merchantibility and fitness for particular purpose, with respect to the hardware, the accompanying product's manual(s) and written materials, and any accompanying hardware. This limited warranty gives you specific legal rights.

Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.

1.7 Packing List

Before you begin installing your single board, please make sure that the following materials have been shipped:

1 x ETX-742E ETX® CPU Module

1 x Driver CD

1 x Quick Installation Guide

If any of the above items is damaged or missing, contact your vendor immediately.

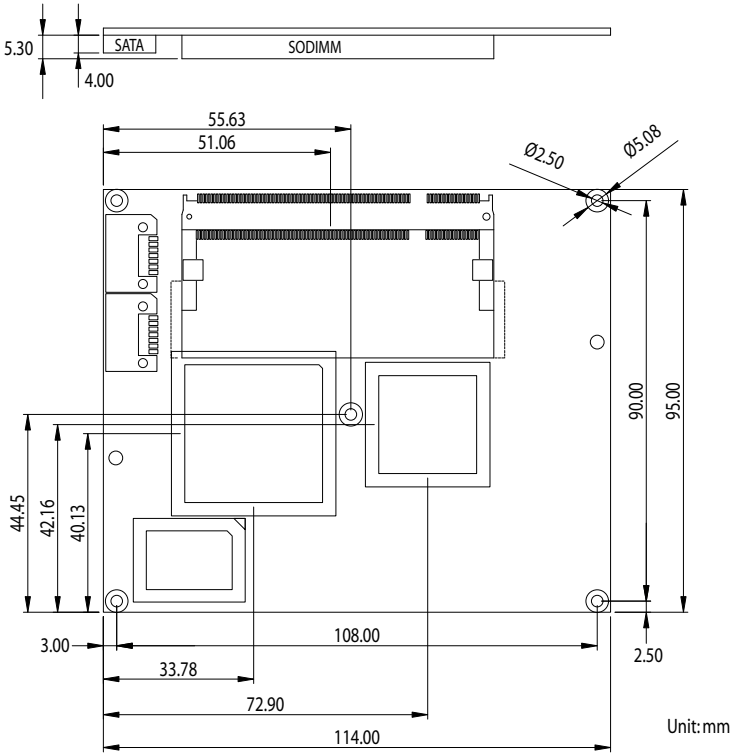
1.8 Ordering Information

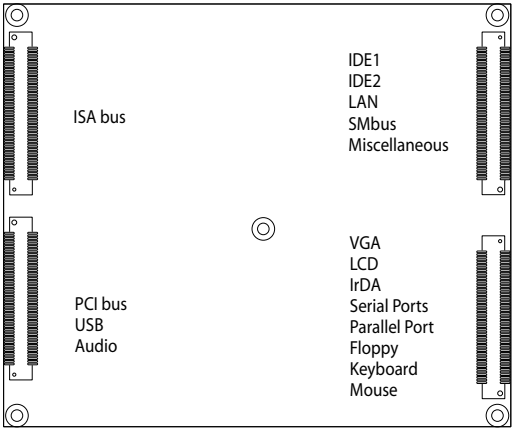
ETX-742E	Intel® Atom N450 1.6GHz ETX® CPU Module
HS-0742-F2	Heat spreader 114 x 95 x 18mm
PBE-1000 R2.1	ETX® evaluation board in ATX form factor
CBK-05-1000-00	Cable kit for PBE-1000 R2.1

1.9 Specifications

Form Factor	ETX CPU Module
CPU	Intel® Atom™ N450 at 1.6GHz Processor
Chipset	Intel® ICH8M
System Memory	1 x 200-pin SO-DIMM socket Up to 2GB DDR2 667MHz SDRAM
VGA/ LCD Controller	Integrated Intel® Graphics Media Accelerator 3150 with Analog RGB/ Single Channel 18-bit LVDS
Ethernet	1 x Realtek 8103EL PCIe 10/100 Base-T Ethernet
BIOS	AMI PnP Flash BIOS
Serial ATA	2 x Serial ATA with 300MB/s HDD transfer rate
IDE Interface	2 x Ultra ATA, support 4 IDE devices
Serial Port	2 x COM ports
Parallel Port	1 x SPP/EPP/ECP mode 1 x Floppy connector, shared with Parallel Port #1
KBMS	Supports PS/2 interface Keyboard and Mouse
Universal Serial Bus	4 x USB 2.0 ports
LCD	Single Channel 18-bit LVDS
Expansion Interface	4 x PCI masters ISA Bus LPC interface
Operation Temp.	-40°C ~ 85°C (-40°F~185°F)
Watchdog Timer	1~255 Level Reset
Dimension (L x W)	114 x 95 mm (4.5 ” x 3.7 ”)

1.10 Board Dimensions and Layout





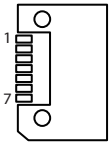
Chapter 2

Installation

2.1 Jumpers and Connectors

SATA1, SATA2 Connectors (Top side)

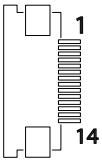
Pin	Description
1	GND
2	TX+
3	TX-
4	GND
5	RX-
6	RX+
7	GND



LPC1 Connector

Connector type: FPC12-14P-P0.5 (Hirose)

Pin	Description
1	LAD0
2	LAD1
3	LAD2
4	LAD3
5	GND
6	LFRAME#
7	INT_SERIRQ
8	BUF_PLT_RST#
9	GND
10	PCLK_CONN
11	GND
12	GND
13	+3.3V
14	+3.3V



ETX1 Connector

A1	GND	GND	A2
A3	PCICLK3	PCICLK4	A4
A5	GND	GND	A6
A7	PCICLK1	PCICLK2	A8
A9	REQ#3	GNT#3	A10
A11	GNT#2	VCC3	A12
A13	REQ#2	GNT#1	A14
A15	REQ#1	VCC3	A16
A17	GNT#0	N.C	A18
A19	VCC	VCC	A20
A21	SERIRQ	REQ#0	A22
A23	AD0	VCC3	A24
A25	AD1	AD2	A26
A27	AD4	AD3	A28
A29	AD6	AD5	A30
A31	CBE#0	AD7	A32
A33	AD8	AD9	A34
A35	GND	GND	A36
A37	AD10	AUXAL	A38
A39	AD11	MIC	A40
A41	AD12	AUXAR	A42
A43	AD13	ASVCC	A44
A45	AD14	SNDL	A46
A47	AD15	ASGND	A48
A49	CBE#1	SNDR	A50
A51	VCC	VCC	A52
A53	PAR	SERR#	A54
A55	PERR#	N.C	A56
A57	PME#	USB2-	A58
A59	LOCK#	DEVSEL#	A60
A61	TRDY#	USB3-	A62
A63	IRDY#	STOP#	A64
A65	FRAME#	USB2+	A66
A67	GND	GND	A68
A69	AD16	CBE#2	A70
A71	AD17	USB3+	A72
A73	AD19	AD18	A74
A75	AD20	USB0-	A76
A77	AD22	AD21	A78
A79	AD23	USB1-	A80
A81	AD24	CBE#3	A82
A83	VCC	VCC	A84
A85	AD25	AD26	A86
A87	AD28	USB0+	A88
A89	AD27	AD29	A90
A91	AD30	USB1+	A92
A93	PCIRST#	AD31	A94
A95	INTR#C	INTR#D	A96
A97	INTR#A	INTR#B	A98
A99	GND	GND	A100

ETX2 Connector

B1	GND	GND	B2
B3	SD14	SD15	B4
B5	SD13	MASTER#	B6
B7	SD12	DREQ7	B8
B9	SD11	DACK#7	B10
B11	SD10	DREQ6	B12
B13	SD9	DACK#6	B14
B15	SD8	DREQ5	B16
B17	MEMW#	DACK#5	B18
B19	MEMR#	DREQ0	B20
B21	LA17	DACK#5	B22
B23	LA18	IRQ14	B24
B25	LA19	IRQ15	B26
B27	LA20	IRQ12	B28
B29	LA21	IRQ11	B30
B31	LA22	IRQ10	B32
B33	LA23	IO16#	B34
B35	GND	GND	B36
B37	SBHE#	M16#	B38
B39	SA0	OSC	B40
B41	SA1	BALE	B42
B43	SA2	TC	B44
B45	SA3	DACK#2	B46
B47	SA4	IRQ3	B48
B49	SA5	IRQ4	B50
B51	VCC	VCC	B52
B53	SA6	IRQ5	B54
B55	SA7	IRQ6	B56
B57	SA8	IRQ7	B58
B59	SA9	SYSCLK	B60
B61	SA10	REFCH#	B62
B63	SA11	DREQ1	B64
B65	SA12	DACK#1	B66
B67	GND	GND	B68
B69	SA13	DREQ3	B70
B71	SA14	DACK#3	B72
B73	SA15	IOR#	B74
B75	SA16	IOW#	B76
B77	SA18	SA17	B78
B79	SA19	SMEMR#	B80
B81	IOCHRDY	AEN	B82
B83	VCC	VCC	B84
B85	SD0	SMEMW#	B86
B87	SD2	SD1	B88
B89	SD3	NOWS#	B90
B91	DREQ2	SD4	B92
B93	SD5	IRQ9	B94
B95	SD9	SD7	B96
B97	IOCHK#	RSTDRV	B98
B99	GND	GND	B100

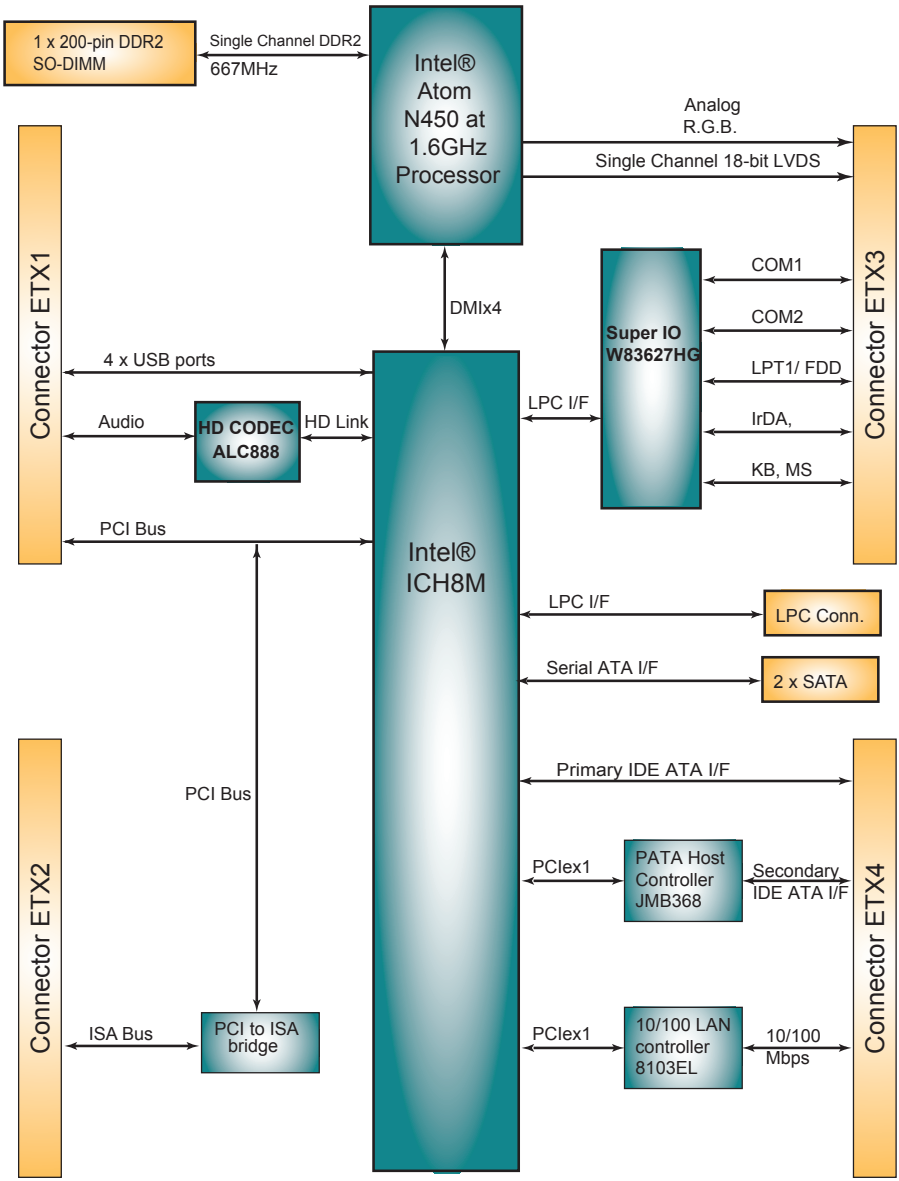
ETX3 Connector

C1	GND	GND	C2
C3	R	B	C4
C5	HSY	G	C6
C7	VSY	Analog RGB_DDC_CLK	C8
C9	DETECT#	Analog RGB_DDC_DATA	C10
C11	TX2CLK#	N.C.	C12
C13	TX2CLK	N.C.	C14
C15	GND	GND	C16
C17	TX2D1	TX2D2	C18
C19	TX2D1#	TX2D2#	C20
C21	GND	GND	C22
C23	N.C.	TX2D0	C24
C25	N.C.	TX2D0#	C26
C27	GND	GND	C28
C29	TX1D2#	TX1CLK	C30
C31	TX1D2	TX1CLK#	C32
C33	GND	GND	C34
C35	TX1D0	TX1D1	C36
C37	TX1D0#	TX1D1#	C38
C39	VCC	VCC	C40
C41	DDC_DATA	N.C.	C42
C43	DDC_CLK	BLON#	C44
C45	BKLTCTL	VDDEN	C46
C47	TV_DATA_COMP	Y	C48
C49	N.C.	C	C50
C51	LPT/FLPY#	N.C.	C52
C53	VCC	GND	C54
C55	STB#	AFD#/DENSEL	C56
C57	N.C.	PD7/N.C	C58
C59	IRRX	ERR#/HDSEL#	C60
C61	IRTX	PD6/N.C	C62
C63	RXD2	INIT#/DIR#	C64
C65	GND	GND	C66
C67	RTS#2	PD5/N.C	C68
C69	DTR#2	SLIN#/STEP#	C70
C71	DCD#2	PD4/DSKCHG#	C72
C73	DSR#2	PD3/RDATA#	C74
C75	CTS#2	PD2/WP#	C76
C77	TXD#2	PD1/TRK0#	C78
C79	RI#2	PD0/INDEX#	C80
C81	VCC	VCC	C82
C83	RXD1	ACK#/DRV	C84
C85	RTS#1	BUSY#/MOT	C86
C87	DTR#1	PE/WDATA#	C88
C89	DCD#1	SLCT#/WGATE#	C90
C91	DSR#1	MSCLK	C92
C93	CTS#1	MSDAT	C94
C95	TXD#1	KBCLK	C96
C97	RI#1	KBDAT	C98
C99	GND	GND	C100

ETX4 Connector

D1	GND	GND	D2
D3	5V_SB	PWGIN	D4
D5	PS_ON	SPEAKER	D6
D7	PWERBTN#	BATT	D8
D9	KBINH	LILED	D10
D11	RSMRST#	ACTLED	D12
D13	N.C	SPEEDLED	D14
D15	N.C	I2CLK	D16
D17	VCC	VCC	D18
D19	OVCR#	N.C	D20
D21	EXTSMI#	I2DAT	D22
D23	SMBCLK	SMBDAT	D24
D25	SIDE_CS1#	SMBALRT#	D26
D27	SIDE_CS0#	SATALED#	D28
D29	SIDE_A2	PIDE_CS3#	D30
D31	SIDE_A0	PIDE_CS1#	D32
D33	GND	GND	D34
D35	PDIAG_S	PIDE_A2	D36
D37	SIDE_A1	PIDE_A0	D38
D39	SIDE_INTRQ	PIDE_A1	D40
D41	BATLOW#	N.C	D42
D43	SIDE_ACK#	PIDE_INTRQ	D44
D45	SIDE_RDY	PIDE_ACK#	D46
D47	SIDE_IOR#	PIDE_RDY	D48
D49	VCC	VCC	D50
D51	SIDE_IOW#	PIDE_IOR#	D52
D53	SIDE_DRQ	PIDE_IOW#	D54
D55	SIDE_D15	PIDE_DRQ	D56
D57	SIDE_D0	PIDE_D15	D58
D59	SIDE_D14	PIDE_D0	D60
D61	SIDE_D1	PIDE_D14	D62
D63	SIDE_D13	PIDE_D1	D64
D65	GND	GND	D66
D67	SIDE_D2	PIDE_D13	D68
D69	SIDE_D12	PIDE_D2	D70
D71	SIDE_D3	PIDE_D12	D72
D73	SIDE_D11	PIDE_D3	D74
D75	SIDE_D4	PIDE_D11	D76
D77	SIDE_D10	PIDE_D4	D78
D79	SIDE_D5	PIDE_D10	D80
D81	VCC	VCC	D82
D83	SIDE_D9	PIDE_D5	D84
D85	SIDE_D6	PIDE_D9	D86
D87	SIDE_D8	PIDE_D6	D88
D89	GPE2#	CBLID_P#	D90
D91	RXD-	PIDE_D8	D92
D93	RXD+	N.C	D94
D95	TXD-	PIDE_D7	D96
D97	TXD+	HDRST#	D98
D99	GND	GND	D100

2.2 Block Diagram



2.3 Driver Installation Paths

Driver	Path
CHIPSET	\CHIPSET\INTEL\INF 9.1
VGA	\GRAPHICS\INTEL_2K_XP_32\5182
AUDIO	\AUDIO\REALTEK_HD\WINDOWS_R198
LAN	\ETHERNET\REALTEK\8103L_WIN5736

Chapter 3

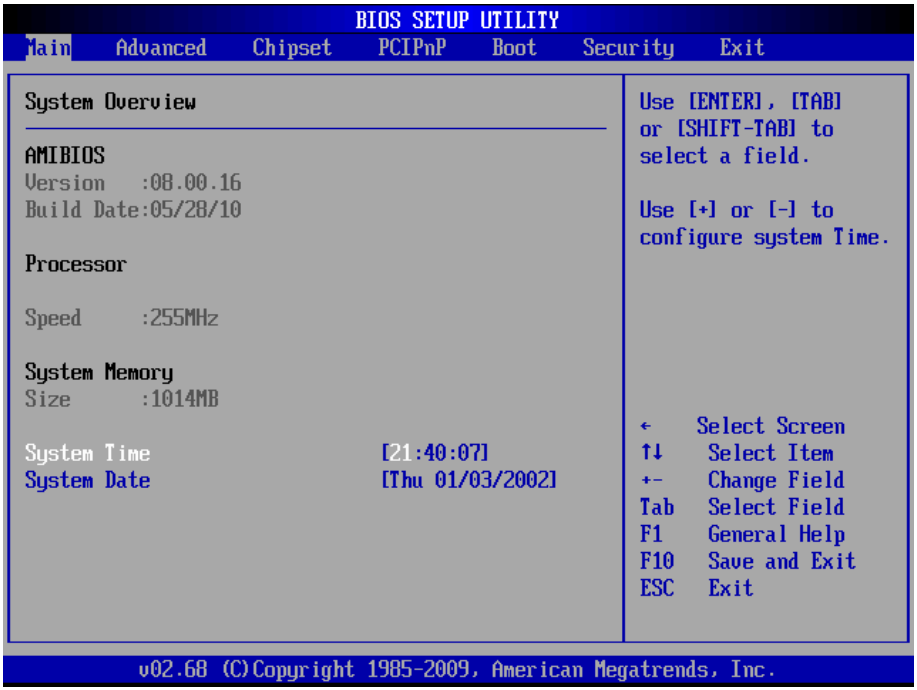
BIOS

3.1 BIOS Main Setup

The AMI BIOS provides a Setup utility program for specifying the system configurations and settings. The BIOS RAM of the system stores the Setup utility and configurations.

When you turn on the computer, the AMI BIOS is immediately activated.
To enter the BIOS SETUP UTILILTY, press **"Delete"** once the power is turned on.
When the computer is shut down, the battery on the motherboard supplies the power for BIOS RAM.

The **Main Setup** screen lists the following information
System Overview
BIOS Version: displays the current version information of the BIOS
Build Date: the date that the BIOS version was made/updated
Processor (auto-detected if installed)
Speed: displays the processor speed
System Memory (auto-detected if installed)
Size: lists the memory size information



Key Commands

BIOS Setup Utility is mainly a key-based navigation interface. Please refer to the following key command instructions for navigation process.

“←”“→”	Move to highlight a particular configuration screen from the top menu bar / Move to highlight items on the screen
“↓” “↑”	Move to highlight previous/next item
Enter	Select and access a setup item/field
Esc:	On the Main Menu – Quit the setup and not save changes into CMOS (a message screen will display and ask you to select “OK” or “Cancel” for exiting and discarding changes. Use “←” and “→” to select and press “Enter” to confirm) On the Sub Menu – Exit current page and return to main menu
Page Up / +	Increase the numeric value on a selected setup item / make change
Page Down -:	Decrease the numeric value on a selected setup item / make change
F1	Activate “General Help” screen
F10:	Save the changes that have been made in the setup and exit. (a message screen will display and ask you to select “OK” or “Cancel” for exiting and saving changes. Use “←” and “→” to select and press “Enter” to confirm)

System Time

Set the system time.

The time format is:
Hour : 00 to 23
Minute : 00 to 59
Second : 00 to 59

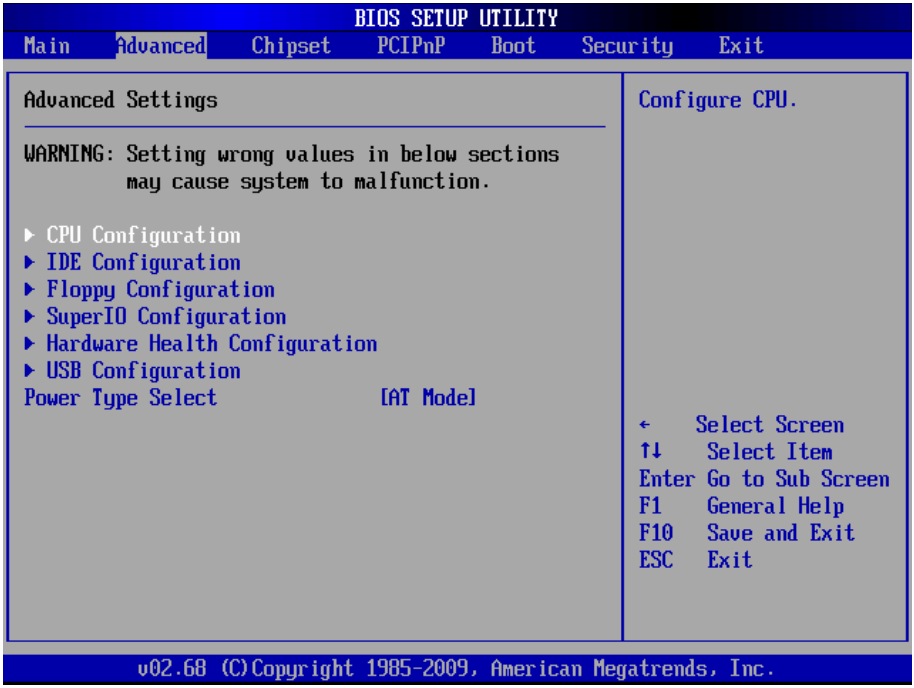
System Date

Set the system date. Note that the ‘Day’ automatically changes when you set the date.

The date format is:
Day : Sun to Sat
Month : 1 to 12
Date : 1 to 31
Year : 1999 to 2099

3.2 Advanced Settings

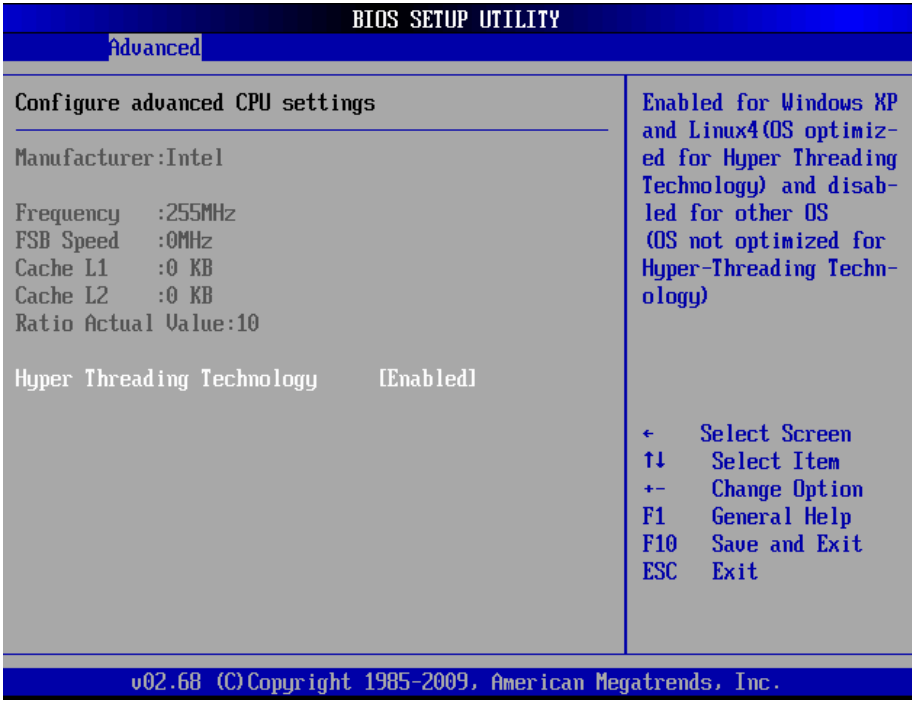
The “Advanced” screen provides the setting options to configure CPU, IDE, Super IO and other peripherals. You can use “←” and “→” keys to select “Advanced” and use the “↓” and “↑” to select a setup item.



Note: please pay attention to the “WARNING” part at the left frame before you decide to configure any setting of an item.

3.2.1 CPU Configuration

Press “Enter” on “CPU Configuration” and you will be able to configure the CPU on the “Configure advanced CPU settings” screen.



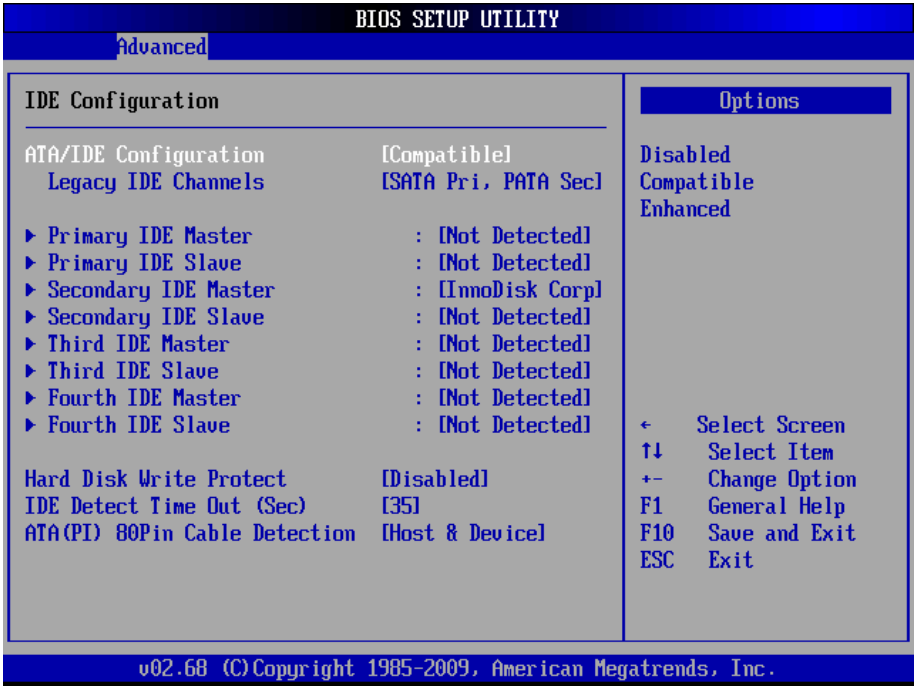
CPU Details
Manufacturer: shows the name of the CPU manufacturer
Frequency: indicates the processor speed
FSB Speed: the data flow speed of FSB (Front Side Bus)
Cache L1: shows the Cache L1 size for the CPU
Cache L2: shows the Cache L2 size for the CPU
Ratio Actual Value: actual value of clock ratio for the CPU

Hyper-Threading Technology

Options
Enabled: Enabled the Hyper-Threading Technology for higher CPU threading speed. (recommended)
Disabled: Disabled the Hyper-Threading Technology.

3.2.2 IDE Configuration

Select the “IDE Configuration to configure the IDE settings. When an item is selected, there is a status description appearing at the right. You can use “Page Up/+” and “Page Down/-” keys to change the value of a selected item.



ATA/IDE Configuration

Configures the options of ATA/IDE controllers connected to the board

Disabled: disables the ATA/IDE controllers connected to the board
Compatible: sets the ATA/IDE controllers to be compatible
Enhanced: sets the ATA/IDE controllers to be in enhanced mode

Legacy IDE Channels (SATA Pri, PATA Sec): specifies SATA or PATA controllers to be primary or secondary.

Primary IDE Master/Slave, Secondary IDE Master/Slave, Third IDE Maser/Slave, Fourth IDE Master/Slave

The BIOS Setup displays all the available, connected IDE devices as well as the IDE status. You may enter a specific IDE device to do particular configurations. Press “Enter” to access the submenu of an IDE device on the list.

Hard Disk Write Protect

Enable or disable Hard Disk Write Protect. If you select “Enabled”, the hard disk will turn into a “write-protected” mode.

IDE Detect Time-out (sec)

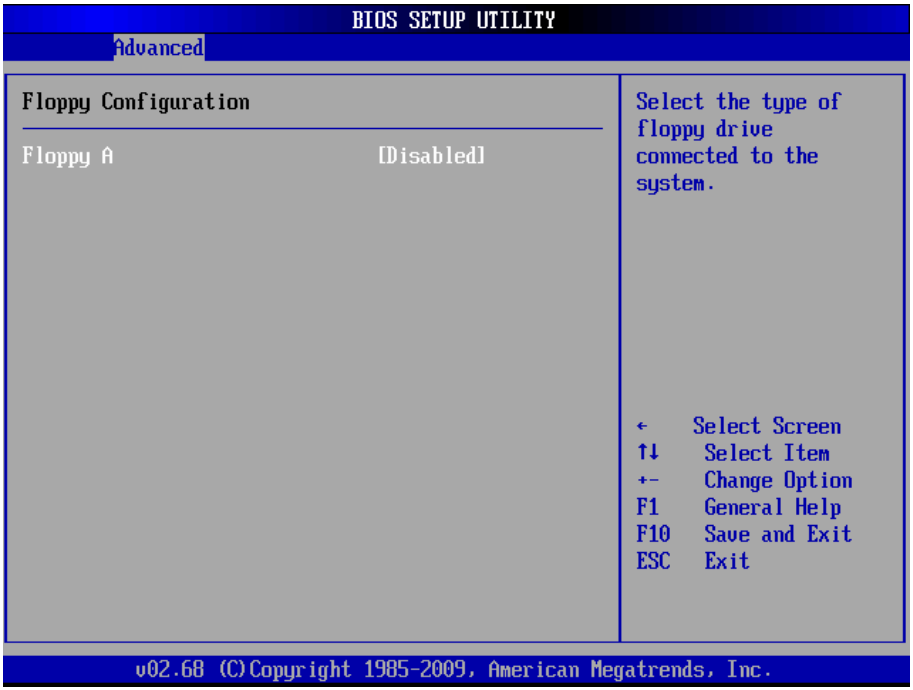
Specifies the delay time for initializing IDE devices. The default value is 0.

ATA (PI) 80Pin Cable Detection

You can set it as “Host & Device”, “Host” or “Device”. Host refers to the capability of IDE controllers to be able to detect connected IDE cable, while Device is defined as the ability of IDE devices to recognize the connected IDE cable.

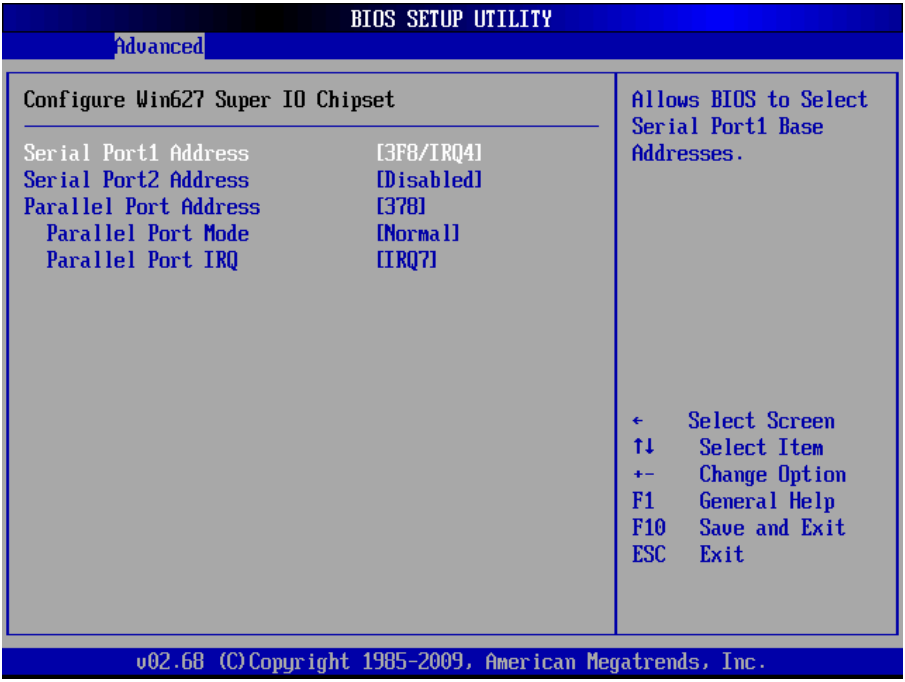
3.2.3 Floppy Configuration

On the “Floppy” screen, you can enable or disable the floppy drive connected to your system.



3.2.4 Super IO Configuration

Use “Super IO Configuration to specify address and modes for Serial Port and Parallel Port.



Serial Port1 / Port2 Address

Select an address and corresponding interrupt for the first and second serial ports.

3F8/IRQ4
2F8/IRQ3
2E8/IRQ3
3E8/IRQ4
Disabled
Auto

Serial Port2 Mode

Allows BIOS to select mode for serial Port2.

Parallel Port Address

Select an address for the parallel port.

- 3BC
- 378
- 278
- Disabled

Parallel Port Mode

Select an operating mode for the onboard parallel port. Select Normal, Compatible or SPP unless you are certain your hardware and software both support one of the other available modes.

- SPP
- EPP
- ECP
- ECP + EPP
- Normal

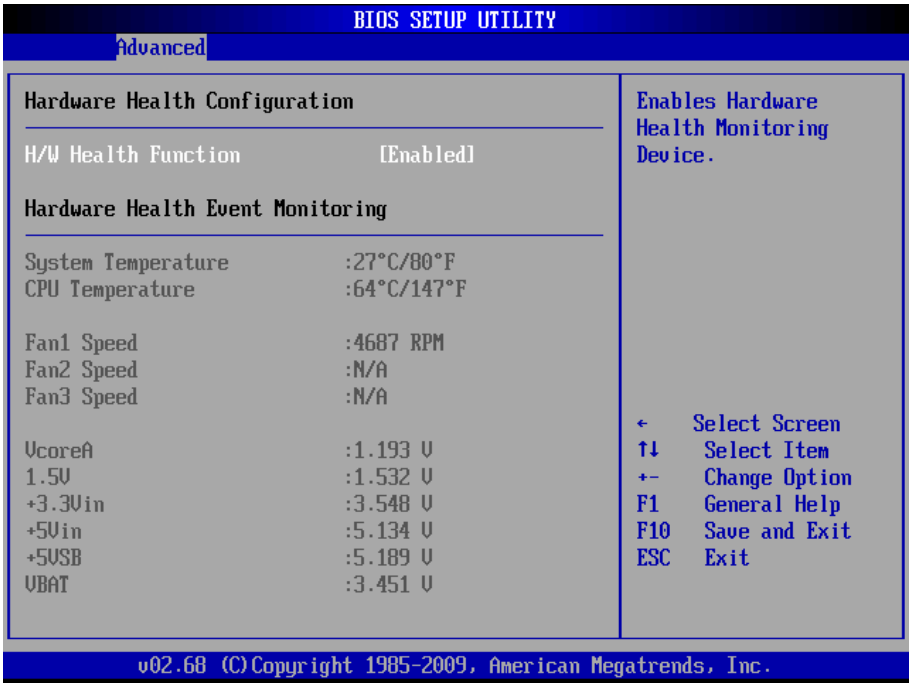
Parallel Port IRQ

Select an interrupt for the parallel port.

- IRQ5
- IRQ7

3.2.5 Hardware Health Configuration

The “Hardware Health Configuration” lists out the temperature and voltage information that is being monitored. The default for “H/W Health Function” is “Enabled.



System Temperature

Show you the currently monitored system temperature.

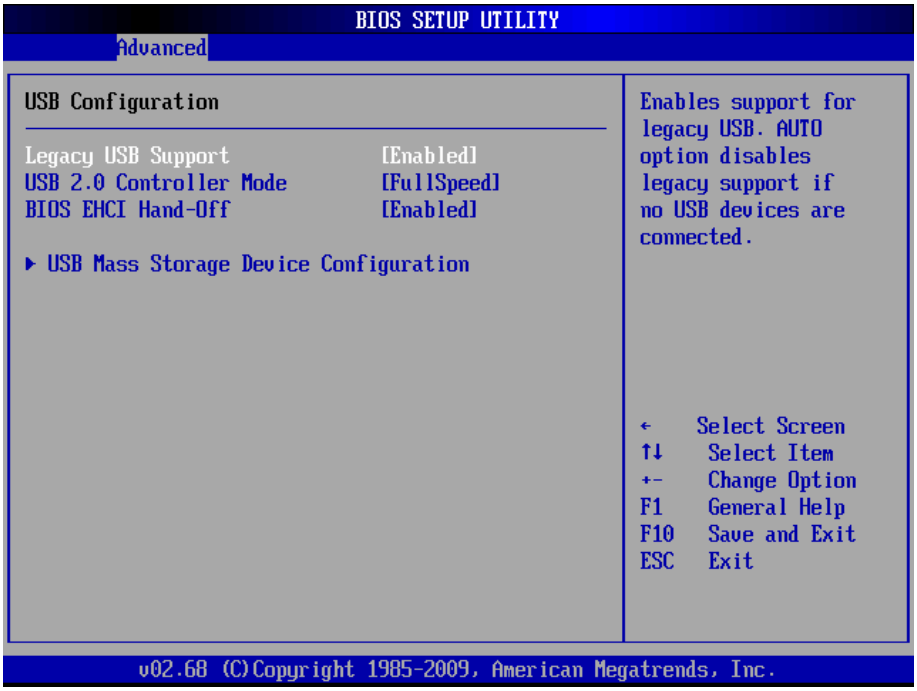
CPU Temperature

Show you the currently monitored CPU temperature.

+1.5V/+3.3Vin / +5Vin / +5VSB/VBAT

Show you the voltage level of the +1.5V, +3.3Vin, +5Vin, +5VSB, or VBAT standby and battery.

3.2.5 USB Configuration



Legacy USB Support

Enables support for legacy USB. AUTO option disables legacy support if no USB devices are connected.

USB 2.0 Controller Mode

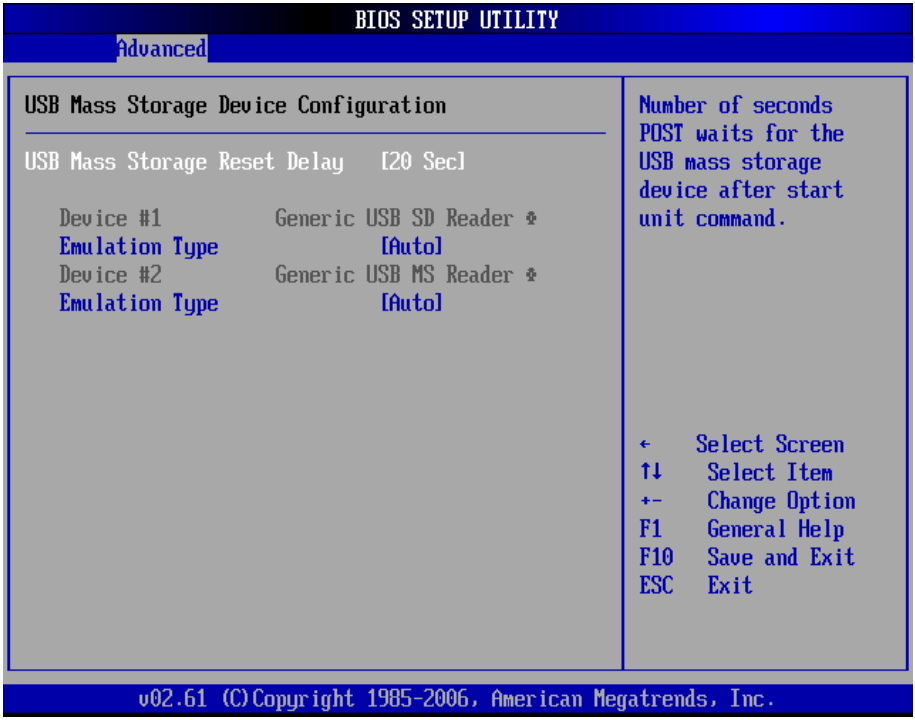
Configures the USB 2.0 controller in High Speed (480Mbps) or Full Speed (12MBPS).

BIOS EHCI Hand-Off

Enabled: enables the EHCI Hand-Off function by BIOS
Disabled: disables the EHCI Hand-Off function by BIOS
Note: this setting potion allows you to enable EHCI Hand Off if your computer operating system does not support it.
EHCI is the abbreviation for Enhanced Host Controller Interface which is necessary for high speed USB operation.

USB Mass Storage Device Configuration

USB Mass Storage Reset Delay:
Number of seconds POST (Power-On Self-Test) waits for the USB mass storage device after start unit command.



Emulation Type

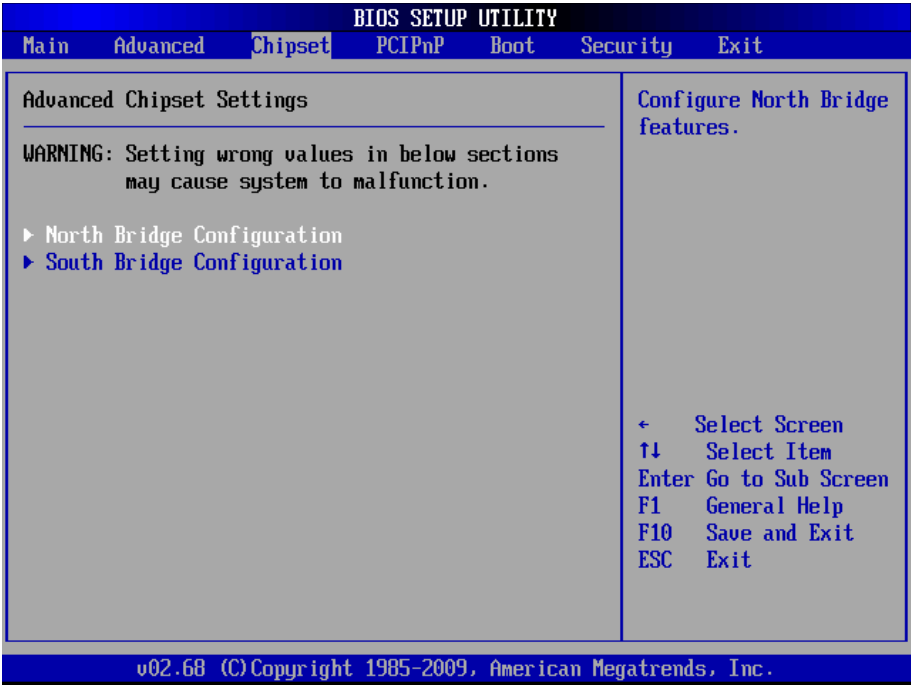
Sets the value for the system to select the emulation type for USB devices. In general, options include “Auto”, “FDD” and “HDD” (HDD stands for Hard Disk Drive, while FDD is also known as 3 1/2 floppy).
Please keep in mind that options such as “FDD” might not always be available as some computers are not built with this type of connectors.

Note

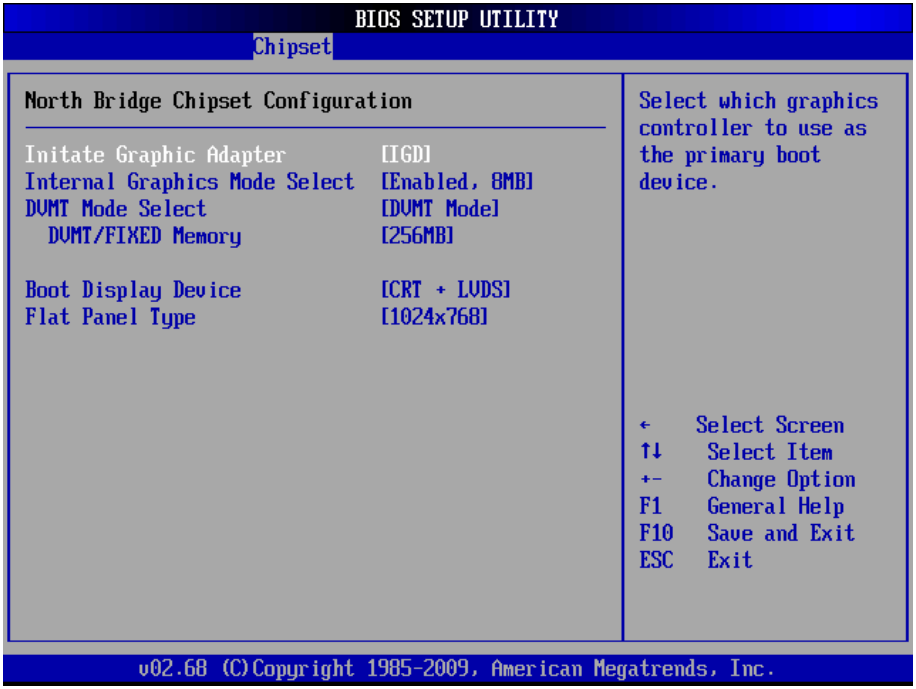
If “Auto” is selected, USB device with storage less than 530MB will be emulated as Floppy and remain as hard drive. Forced FDD option can be used to force a HDD formatted drive to “BOOT” as FDD (for example, ZIP drive)

3.3 Chipset

Select “Chipset” to access to “North Bridge Configuration” and “South Bridge Configuration”. You can enter the sub menu of the two configuration options.



3.3.1 North Bridge Chipset Configuration



Initiate Graphic Adapter:
Selects which graphics controller to be used as the primary boot device.

Internal Graphic Mode Select:
Selects the amount of the system memory to enable the internal graphic mode

DVMT Mode

Setting: FIXED, DVMT (Default), BOTH.

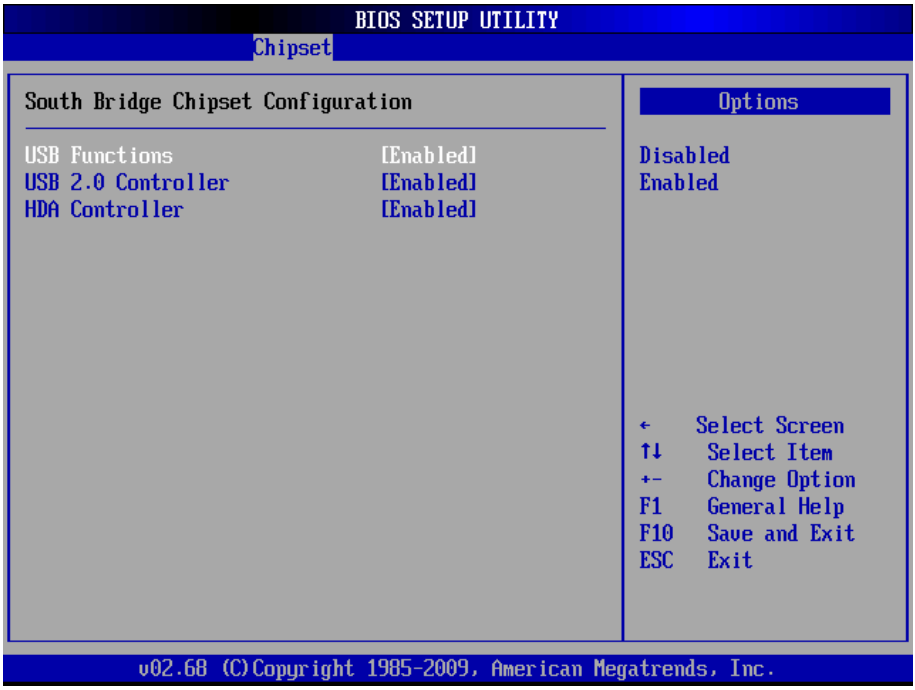
DVMT/FIXED Memory Size

Setting: 64MB, 128MB (Default), 224MB.

Boot Display Device: boot setting for the display device connected to the computer, such as “External CRT” monitor.
Flat Panel Type: the resolution types of the connected flat panel display device.

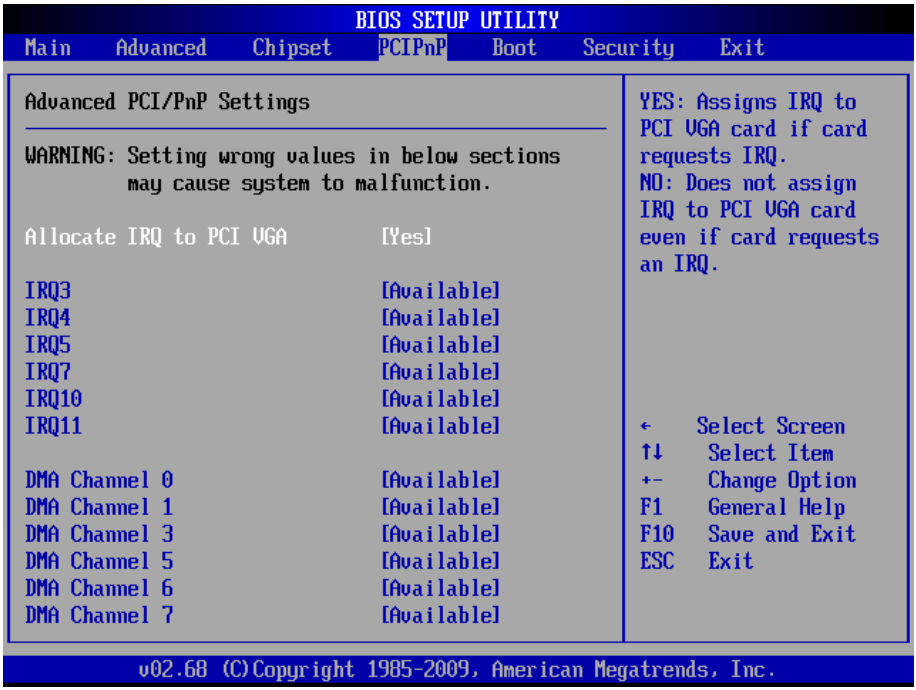
3.3.2 South Bridge Chipset Configuration

Normally, the south bridge controls the basic I/O functions, such as USB and audio. This screen allows you to access to the configurations of the I/Os.



3.4 PCIPnP

The “PCIPnP” screen provides advanced setting options for your PCI or PnP (plug and play) peripherals.



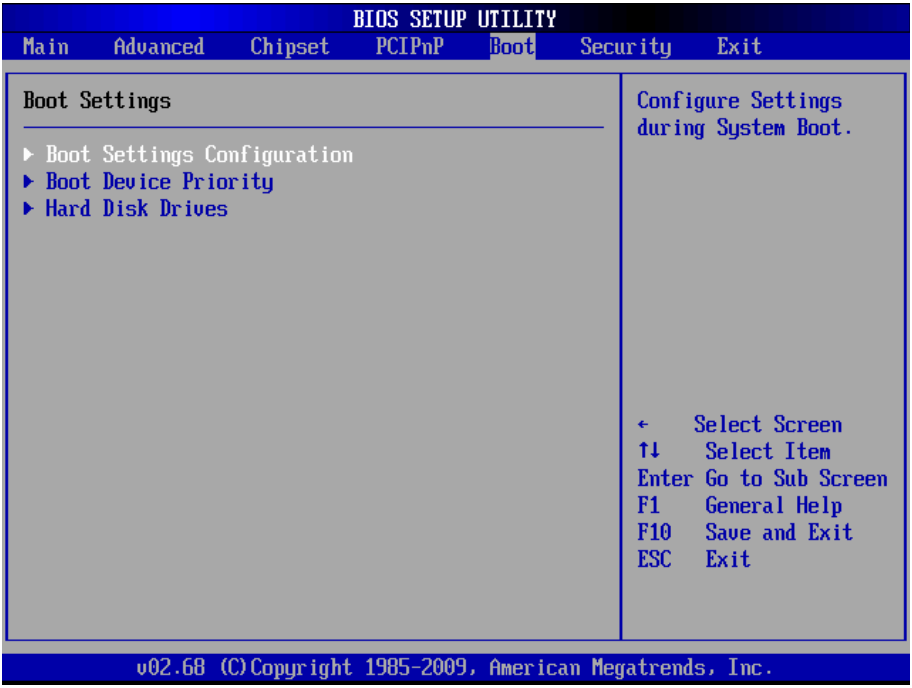
Allocate IRQ to PCI VGA:
[Yes]: assigns IRQ to PCI VGA card if card requests IRQ
[No]: does not assign IRQ to PCI VGA card even if card requests IRQ

[Available]: if an item is specified “Available”, the particular item can be used by PCI or PnP peripherals/devices
[Reserved]: if an item is specified as “Reserved”, the particular item can only be used by legacy ISA peripherals/devices

Note: please pay attention to the “WARNING” part at the left frame before you decide to configure any setting of an item.

3.5 Boot

The “Boot” screen provides the access to configure the settings for system boot.

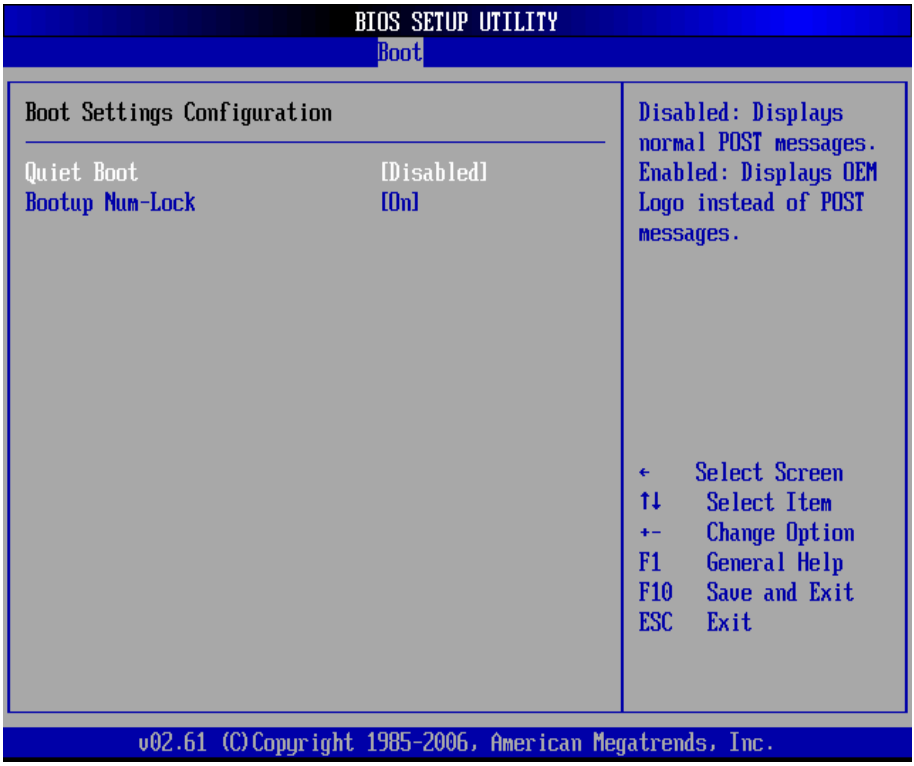


Boot Setting Configuration: enter the sub menu for boot setting.

Boot Device Priority: access to the sub menu for boot device priority.

Hard Disk Drives: configure the boot settings for the Hard Disk Drives connected to the system.

3.5.1 Boot Setting Configuration

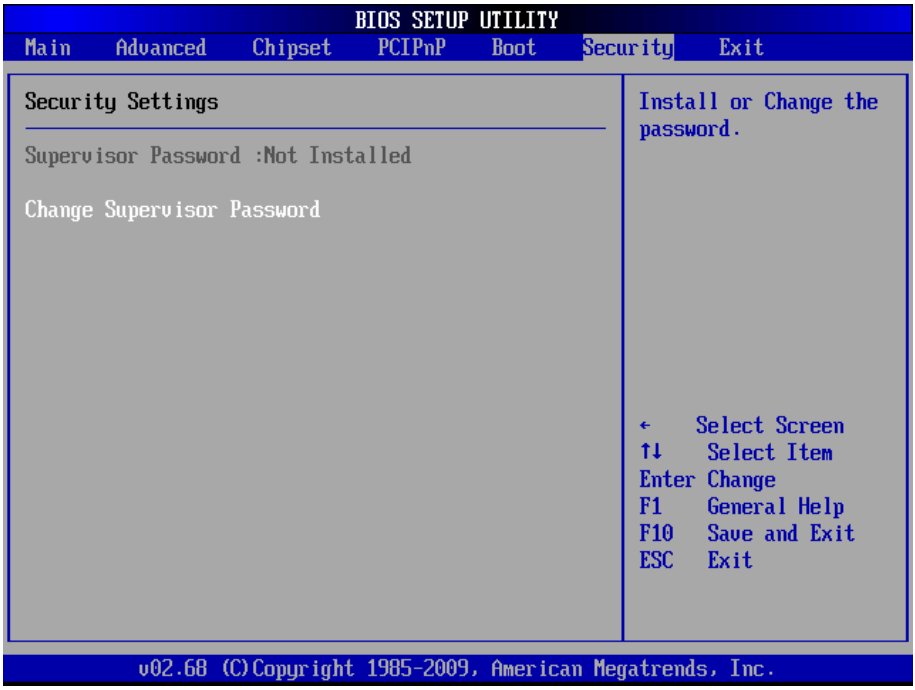


Quiet Boot: displays normal POST messages when it's selected as "Disabled". When it is set as "Enabled", OEM messages will be displayed instead of POST messages. The default is "Disabled".

Bootup Num-Lock: modifies Number Lock setting when the system boots up. Select "On" to automatically enable the Number Lock on keyboard when the system is booting up.

3.6 Security

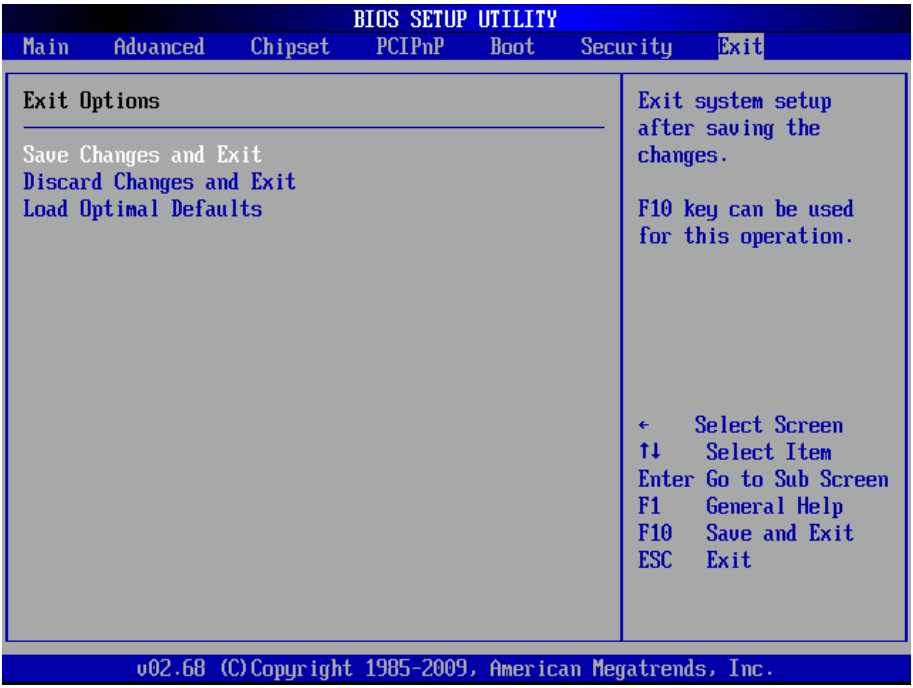
The “Security Settings” screen allows you to set password.



Change Supervisor Password: the default is “Not Installed”, but you can change the Supervisor Password and then it will appear “Installed”. Please always remember your password or else you will have to reset the whole system.

3.7 Exit

Select “Exit” to set exit options, save changes or load default values.



Save Changes and Exit

When you press “Enter” on this option, a message described as the one below will appear:

“Save configuration changes and exit setup?”

Pressing <OK> stores the configuration changes made in BIOS in CMOS menu - a special section of memory that stays on after you turn your system off, and then exit. The next time you boot your system up, the new configured system values will take place.

Note: you can also press <F10> to enable this operation.

Discard Changes and Exit

Exit system setup without saving any changes.
You can also press <ESC> to activate this function.

Load Optimal Defaults

When you press <Enter> on this option, a message dialog box will appear asking for your confirmation:

Load Optimal Defaults?
[OK] [Cancel]

Press [OK] to load the BIOS Optimal Default values for all the setup options.

You can also press <F9> key to enable this operation.

3.8 AMI BIOS Checkpoints

3.8.1 Bootblock Initialization Code Checkpoints

The Bootblock initialization code sets up the chipset, memory and other components before system memory is available. The following table describes the type of checkpoints that may occur during the bootblock initialization portion of the BIOS *(Note)*:

Checkpoint	Description
Before D0	If boot block debugger is enabled, CPU cache-as-RAM functionality is enabled at this point. Stack will be enabled from this point.
D0	Early Boot Strap Processo (BSP) initialization like microcode update, frequency and other CPU cirtical initialization. Early chipset initialization is done.
D1	Early super I/O initialization is done including RTC and keyboard controller. Serial port is enabled at this point if needed for debugging. NMI is deisabled. Perfrom keyboard controller BAT test. Save power-on CPUID value in scretch CMOS. Go to flat mode with 4GB limit and GA20 enabled.
D2	Verify the boot block checksum. System will hang here if checksum is bad.
D3	Disable CACHE before memory detection. Execute full memory sizing module. If memory sizing module not executed, start memory refresh and do memory sizing in Boot block code. Do additional chipset initialization. Re-enabled CACHE. Verify that flat mode is enabled.
D4	Test base 512KB memory. Adjust policies and cache first 8MB. Set stack.
D5	Bootblock code is copied from ROM to lower system memory and control is given to it. BIOS now executes out of RAM. Copies compressed boot block code to memory in right segments. Copies BIOS from ROM to RAM for faster access. Perfroms main BIOS checksum and updates recovery status accordingly.

D6	Both key sequence and OEM specific method is checked to determine if BIOS recovery is forced. If BIOS recovery is necessary, control flows tocheckpoint E0. Seed *Bootblock Recovery Code Checkpoints* section of document for more information.
D7	Restore CPUID value back into register. The Bootblock-Runtime interface module is moved to system memory and control is given to it. Determine whether in memory.
D8	The Tuntime module is uncompressed into memory. CPUID information is stored in memory.
D9	Store the Uncompressed pointer for future use in PMM. Copying Main BIOS into memory. Leaves all RAM below 1MB Read-Write including E000 and F000 shadow areas but closing SMRAM.
DA	Restore CPUID value back into register. Give control to BIOS POS (ExecutePOSTKernel). See *POST Code Checkpoints* section of document for more information.
DC	System is saking from ACPI S3 state.
E1 - E8 EC - EE	OEM memory detection / configuration error. This range is reserved for chipset vendors & system manufacturers. The error associated with this value may be different from one platform to be next.

3.8.2 Bootclock Recovery Code Checkpoints

The Bootblock recovery code gets control when the BIOS determines that a BIOS recovery needs to occur because the user has forced the update or the BIOS checksum is corrupt. The following table describes the type of checkpoints that may occur during the Bootblock recovery portion of the BIOS *(Note)*:

Checkpoint	Description
E0	Initialize the floppy controller in the super I/O. Some interrupt vectors are initialized. DMA controller is initialized. 8259 interrupt controller is initialized. L2 cache is enabled.
E9	Set up floppy controller and data. Attempt to red from floppy.
EA	Enable ATAPI hardware. Attempt to read from ARMD and ATAPI CDROM.
EB	Disable ATAPI hardware. Jump back to checkpoint E9.
EF	Read error occurred on media. Jump back to checkpoint EB.
F0	Search for pre-defined recovery file name in root directory.
F1	Recovery file not found.
F2	Start reading FAT table and analyze FAT to find the clusters occupied by the recovery file.
F3	Start reading the recovery file cluster by cluster.
F5	Disable L1 cache.
FA	Check the validity of the recovery file configuration to the current configuration of the flash part.
FB	Make flash write enabled through chipset and OEM specific method. Detect proper flash part. Verify that the found flash part size equals the recovery file size.
F4	The recovery file size does not equal the found flash part size.

FC	Erase the flash part.
FD	Program the flash part.
FF	The flash has been updated successfully. Make flash write disabled. Disable ATAPI hardware. Restore CPUID value back into register. Give control to F000 ROM at F000:FFF0h.

3.8.3 POST Code Checkpoints

The POST code checkpoints are the largest set of checkpoints during the BIOS pre-boot process. The following table describes the type of checkpoints that may occur during the POST portion of the BIOS *(Note)*:

Checkpoint	Description
03	Disable NMI, Parity, video for EGA, and DMA controllers. Initialize BIOS, POST, Runtime data area. Also initialize BIOS modules on POST entry and GPNV area. Initialized CMOS as mentioned in the Kernel Variable "wCMOSFlags."
04	Check CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK. Verify CMOS checksum manually by reading storage area. If the CMOS checksum is bad, update CMOS with power-on default values and clear passwords. Initialize status register A. Initializes data variables that are based on CMOS setup questions. Initializes both the 8259 compatible PICs in the system.
05	Initializes the interrupt controlling hardware (generally PIC) and interrupt vector table.
06	Do R/W test to CH-2 count reg. Initialize CH-0 as system timer. Install the POSTINT1Ch handler. Enable IRQ-0 in PIC for system timer interrupt. Traps INT1Ch vector to "POSTINT1ChHandlerBlock."
07	Fixes CPU POST interface calling pointer.
08	Initializes the CPU. The BAT test is being done on KBC. Program the keyboard controller command byte is being done after Auto detection of KB/MS using AMI KB-5.
C0	Early CPU Init Start -- Disable Cache - Init Local APIC
C1	Set up boot strap processor Information
C2	Set up boot strap processor for POST
C5	Enumerate and set up application processors
C6	Re-enable cache for boot strap processor

C7	Early CPU Init Exit
0A	Initializes the 8042 compatible Key Board Controller.
0B	Detects the presence of PS/2 mouse.
0C	Detects the presence of Keyboard in KBC port.
0E	Testing and initialization of different Input Devices. Also, update the Kernel Variables. Traps the INT09h vector, so that the POST INT09h handler gets control for IRQ1. Uncompress all available language, BIOS logo, and Silent logo modules.
13	Early POST initialization of chipset registers.
20	Relocate System Management Interrupt vector for all CPU in the system.
24	Uncompress and initialize any platform specific BIOS modules. GPNV is initialized at this checkpoint.
2A	Initializes different devices through DIM. See DIM Code Checkpoints section of document for more information.
2C	Initializes different devices. Detects and initializes the video adapter installed in the system that have optional ROMs.
2E	Initializes all the output devices.
31	Allocate memory for ADM module and uncompress it. Give control to ADM module for initialization. Initialize language and font modules for ADM. Activate ADM module.
33	Initializes the silent boot module. Set the window for displaying text information.
37	Displaying sign-on message, CPU information, setup key message, and any OEM specific information.

38	Initializes different devices through DIM. See DIM Code Checkpoints section of document for more information. USB controllers are initialized at this point.
39	Initializes DMAC-1 & DMAC-2.
3A	Initialize RTC date/time.
3B	Test for total memory installed in the system. Also, Check for DEL keys to limit memory test. Display total memory in the system.
3C	Mid POST initialization of chipset registers.
40	Detect different devices (Parallel ports, serial ports, and coprocessor in CPU, … etc.) successfully installed in the system and update the BDA, EBDA…etc.
52	Updates CMOS memory size from memory found in memory test. Allocates memory for Extended BIOS Data Area from base memory. Programming the memory hole or any kind of implementation that needs an adjustment in system RAM size if needed.
60	Initializes NUM-LOCK status and programs the KBD typematic rate.
75	Initialize Int-13 and prepare for IPL detection.
78	Initializes IPL devices controlled by BIOS and option ROMs.
7C	Generate and write contents of ESCD in NVRam.
84	Log errors encountered during POST.
85	Display errors to theuser and gets the user response for error.
87	Execute BIOS setup if needed / requested. Check boot password if installed.
8C	Late POST initialization of chipset registers.
8D	Build ACPI tables (if ACPI is supported)
8E	Program the peripheral parameters. Enable/Disalbe NMI as selected.
90	Initialization of system management interrupt by invoking all handlers.
A1	Lian-up work needed before booting to OS.

A2	Takes care of runtime image preparation for different BIOS modules. Fill the free area in F000h segment with 0FFh. Initializes the Microsoft IRQ Routing Table. Prepares the runtime language module. Disables the system configuration display if needed.
A4	Initialize runtime language module. Display boot option popup menu.
A7	Displays the system configuration screen if enabled. Initialize the CPU's before boot, which includes the programming of the MTRR's.
A9	Wait for userinput at config display if needed.
AA	Uninstall POST INT1Ch vector and INT09h vector.
AB	Prepare BBS for Int 19 boot. Init MP tables.
AC	End of POST initialization of chipset registers. De-initializes the ADM module.
B1	Save system context for ACPI. Prepare CPU for OS boot including final MTRR values.
00	Passes control to OS Loader (typically INT19h).

3.8.4 DIM Code Checkpoints

The Device Initialization Manager (DIM) gets control at various times during BIOS POST to initialize different system buses. The following table describes the main checkpoints where the DIM module is accessed *(Note)*:

Checkpoint	Description
2A	Initialize different buses and perform the following functions: Reset, Detect, and Disable (function 0); Static Device Initialization (function); Boot Output Device Initialization (function 2). Function 0 disables all device nodes, PCI devices, and PnP ISA cards. It also assigns PCI bus numbers. Function 1 initializes all static devices that include manual configured onboard peripherals, memory and I/O decode windows in PCI-PCI bridges, and noncompliant PCI devices. Static resources are also reserved. Function 2 searches for and initializes any PnP, PCI, or AGP video devices.
38	Initialize different buses and perform the following functions: Boot Input Device Initialization (function 3); IPL Device Initialization (function 4); General Device Initialization (function 5). Function 3 searches for and configures PCI input devices and detects if system has standard keyboard controller. Function 4 searches for and configures all PnP and PCI boot devices. Function 5 configures all onboard peripherals that are set to an automatic configuration and configures all remaining PnP and PCI devices.

While control is in the different functions, additional checkpoints are output to port 80h as a word value to identify the routines under execution. The low byte value indicates the main POST Code Checkpoint. The high byte is divided into two nibbles and contains two fields. The details of the high byte of these checkpoints are as follows:

HIGH BYTE XY

The upper nibble “X” indicates the function number that is being executed. “X” can be from 0 to 7.

0 = func#0, disable all devices on the BUS concerned.
2 = func#2, output device initialization on the BUS concerned.
3 = func#3, input device initialization on the BUS concerned.
4 = func#4, IPL device initialization on the BUS concerned.
5 = func#5, general device initialization on the BUS concerned.
6 = func#6, error reporting for the BUS concerned.
7 = func#7, add-on ROM initialization for all BUSes.
8 = func#8, BBS ROM initialization for all BUSes.

The lower nibble 'Y' indicates the BUS on which the different routines are being executed. 'Y' can be from 0 to 5.

0 = Generic DIM (Device Initialization Manager).
1 = On-board System devices.
2 = ISA devices.
3 = EISA devices.
4 = ISA PnP devices.
5 = PCI devices.

3.8.5 ACPI Runtime Checkpoints

ACPI checkpoints are displayed when an ACPI capable operating system either enters or leaves a sleep state. The following table describes the type of checkpoints that may occur during ACPI sleep or wake events *(Note)*:

Checkpoint	Description
AC	First ASL check point. Indicates the system is running in ACPI mode.
AA	System is running in APIC mode.
01, 02, 03, 04, 05	Entering sleep state S1, S2, S3, S4, or S5.
10, 20, 30, 40, 50	Waking from sleep state S1, S2, S3, S4, or S5.

Note:
Please note that checkpoints may differ between different platforms based on system configuration. Checkpoints may change due to vendor requirements, system chipset or option ROMs from add-in PCI devices.

This page is intentionally left blank.

Appendix

Appendix A: I/O Port Address Map

Each peripheral device in the system is assigned a set of I/O port addresses which also becomes the identity of the device.
The following table lists the I/O port addresses used.

Address	Device Description
00000000 - 00000007	DMA Controller
00000000 - 00000CF7	PCI bus
00000010 - 0000001F	Motherboard Resource
00000020 - 00000021	Programmable Interrupt Controller
00000022 - 0000003F	Motherboard Resource
00000040 - 00000043	System Timer
00000044 - 0000005F	Motherboard Resource
00000060 - 00000060	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
00000061 - 00000061	System Speaker
00000062 - 00000063	Motherboard Resource
00000064 - 00000064	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
00000065 - 0000006F	Motherboard Resource
00000070 - 00000073	System CMOS/real time clock
00000074 - 0000007F	Motherboard Resource
00000080 - 00000090	DMA Controller
00000091 - 00000093	Motherboard Resource
00000094 - 0000009F	DMA Controller
000000A0 - 000000A1	Programmable Interrupt Controller
000000A2 - 000000BF	Motherboard Resource
000000C0 - 000000DF	DMA Controller
000000E0 - 000000EF	Motherboard Resource
000000F0 - 000000FF	Numeric Data Processor
000001F0 - 000001F7	Primary IDE Channel
00000274 - 00000277	ISAPNP Read Data Port

00000279 - 00000279	ISAPNP Read Data Port
00000294 - 00000297	Motherboard Resource
000002E8 - 000002EF	Communications Port (COM4)
000002F8 - 000002FF	Communications Port (COM2)
00000378 - 0000037F	Printer Port (LPT1)
000003B0 - 000003BB	Mobile Intel® 945 Express Chipset Family
000003C0 - 000003DF	Mobile Intel® 945 Express Chipset Family
000003E8 - 000003EF	Communications Port (COM3)
000003F6 - 000003F6	Primary IDE Channel
000003F8 - 000003FF	Communications Port (COM1)
00000400 - 000004BF	Motherboard Resource
000004D0 - 000004D1	Motherboard Resource
00000500 - 0000051F	Intel® 82801G (ICH7 Family) SMBus Controller - 27DA
00000680 - 000006FF	Motherboard Resource
00000778 - 0000077B	Printer Port (LPT1)
00000880 - 0000088F	Motherboard Resource
00000A78 - 00000A7B	Motherboard Resource
00000BBC - 00000BBF	Motherboard Resource
00000BBC - 00000BBF	Motherboard Resource
00000D00 - 0000FFFF	PCI bus
00000E78 - 00000E7B	Motherboard Resource
00000F78 - 00000F7B	Motherboard Resource
00000FBC - 00000FBF	Motherboard Resource
0000B000 - 0000BFFF	Intel® 82801G (ICH7 Family) PCI Express Root Port - 27D4
0000C000 - 0000CFFF	Intel® 82801G (ICH7 Family) PCI Express Root Port - 27D0
0000DF00 - 0000DF3F	Intel® PRO/100 VE Network Connection
0000F000 - 0000F0FF	Realtek AC'97 Audio
0000F300 - 0000F30F	Intel® 82801GBM/GHM (ICH7-M Family) Serial ATA Storage Controller - 27C4

0000F400 - 0000F40F	Intel® 82801GBM/GHM (ICH7-M Family) Serial ATA Storage Controller - 27C4
0000F500 - 0000F50F	Intel® 82801GBM/GHM (ICH7-M Family) Serial ATA Storage Controller - 27C4

Appendix B: Interrupt Request Lines (IRQ)

Peripheral devices use interrupt request lines to notify CPU for the service required. The following table shows the IRQ used by the devices on board.

Level	Function
IRQ 01	Standard 101/102-Key or Microsoft Natural PS/2 Keyboard
IRQ 03	Communications Port
IRQ 04	Communications Port
IRQ 08	System CMOS/real time clock
IRQ 09	Microsoft ACPI-Compliant System
IRQ 10	Communications Port
IRQ 11	Communications Port
IRQ 12	PS/2 Compatible Mouse
IRQ 13	Math Coprocessor
IRQ 14	Primary IDE Channel
IRQ 15	Intel® 82801G (ICH7 Family) SMBus Controller - 27DA
IRQ 16	Intel® 82801G (ICH7 Family PCI Express Root Port - 27D0
IRQ 16	Intel® 82801G (ICH7 Family) USB Universal Host Controller - 27CB
IRQ 16	Mobile Intel 945GM Express Chipset Family
IRQ 17	Realtek AC'97 Audio
IRQ 18	Intel® 82801G (ICH7 Family) PCI Express Root Port - 27D4
IRQ 18	Intel® 82801G (ICH7 Family) USB Universal Host Controller - 27CA
IRQ 19	Intel® 82801G (ICH7 Family) USB Universal Host Controller - 27C9
IRQ 19	Intel® 82801G (ICH7-M Family) Serial ATA Storage Controller - 27C4
IRQ 19	Intel® 82801G (ICH7 Family) USB Universal Host Controller - 27C9
IRQ 20	Intel® PRO/100 VE Network Connection
IRQ 23	Intel® 82801G (ICH7 Family) USB Universal Host Controller - 27C8

IRQ 23	Intel® 82801G (ICH7 Family) USB2 Enhanced Host Controller - 27CC

Appendix C: Watchdog Timer (WDT) Setting

WDT is widely used for industry application to monitoring the activity of CPU. Application software depends on its requirement to trigger WDT with adequate timer setting. Before WDT time out, the functional normal system will reload the WDT. The WDT never time out for a normal system. Then, WDT will time out and reset the system automatically to avoid abnormal operation.

This board supports 255 levels watchdog timer by software programming. Below are the source codes written in assembly & C, please take them for WDT application examples.

Assembly Code

```
;-- Initial W83627 --
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 87h
        out     DX, AX          ;
        out     DX, AX          ; initial W83627 start
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 2Bh
        out     DX, AL          ; Select CR2B
        mov     AL, 00h
        inc     DX
        out     DX, AL          ; Set CR2B bit 4=0, PIN89=WDTO
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 07h
        out     DX, AL          ; Point to Logical Device Selector
        mov     AL, 08h
        inc     DX
        out     DX, AL          ; Select Logical Device 8
;--
```

```
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 30h
        out     DX, AL          ; select CR30
        mov     AL, 01h
        inc     DX
        out     DX, AL          ; update CR30 to 01h
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F0h
        out     DX, AL          ; select CRF0
        mov     AL, 00h
        inc     DX
        out     DX, AL          ; set CRF0=00h, output
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F5h
        out     DX, AL          ; select CRF5, WDT Timer unit
        mov     AL, 00h         ; bit2 =0 ->second ; bit2 =1 -> minute
        inc     DX
        out     DX, AL          ; update CRF5 bit2 to 00h
;--
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, 0F6h
        out     DX, AL          ; select CRF6, WDT Timer
        mov     AL, 05h
        inc     DX
        out     DX, AL          ; update CRF6 to 5 unit
;---
        mov     AX, 2Eh
        mov     DX, AX
        mov     AL, AAh
        out     DX, AX
;-- end
```

C language Code

```
/*-----     Include Header Area -----*/
#include "math.h"
#include "stdio.h"
#include "dos.h"

/*-----     routing, sub-routing -----*/

void main()

{
          outportb(0x2e, 0x87);       /* initial IO port twice */
          outportb(0x2e, 0x87);

          outportb(0x2e, 0x2B);       /* select CR2B */
          outportb(0x2e+1, 0x00);     /* update CR2B bit4 to 00h */
                                      /* Set PIN89 as WDTO */

          outportb(0x2e, 0x07);       /* point to logical device selector */
          outportb(0x2e+1, 0x08);     /* select logical device 8 */
          outportb(0x2e, 0x30);       /* select CR30 */
          outportb(0x2e+1, 0x01);     /* update CR30 to 01h */
          outportb(0x2e, 0xf0);       /* select CRF0 */
          outportb(0x2e+1, 0x00);     /* update CRF0 to 00h */
          outportb(0x2e, 0xf5);       /* select CRF5 to set timer unit */
          outportb(0x2e+1, 0x00);     /* update CRF5 bit2, 0:sec; 1:Min. */
          outportb(0x2e, 0xF6);       /* select CRF6 */
          outportb(0x2e+1, 0x05);     /* update CRF6 to 05h (5 sec) */

          outportb(0x2e, 0xAA);       /* stop program W83627, Exit */
}
```